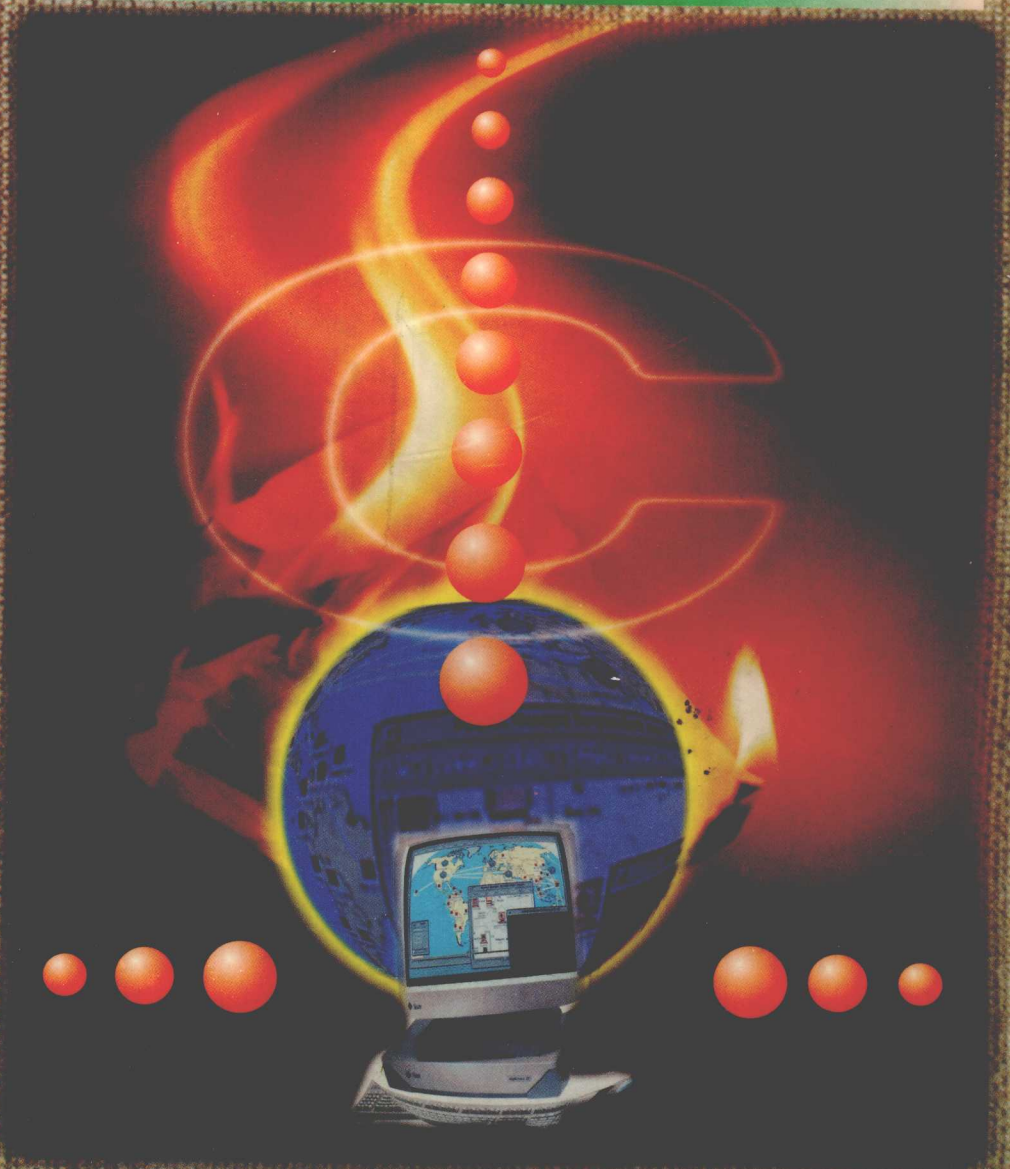
高校计算机教材丛书

C语言程序设计

（修订本）

齐勇 冯博琴 王建仁



COMPUTER

西安交通大学出版社

高校计算机教材丛书

C 语言程序设计

（修订本）

齐 勇 冯博琴 王建仁

西安交通大学出版社

内 容 简 介

本书详细地介绍了C语言的基础知识和程序设计。主要内容包括数据定义,运算符及表达式,语句及流程控制,函数,输入/输出,屏幕管理,菜单和图形设计,操作系统对C语言的支撑,C程序的动态调试,同时介绍了面向对象技术和C++。

本书结构新颖,内容朴实。正面回答了C有哪些特点,为什么有这些特点,与其他语言有何不同,有何利弊?作者精选了丰富的例子,并均在微机上通过。考虑到读者的特点,概念阐述清楚,结论有根有据。

本书可作为普通高校有关专业的教材或供计算机应用人员参考。成教和职业培训也可选用它作为参考书。

(陕)新登字007号

高校计算机教材丛书
C语言程序设计
(修订本)
齐 勇 冯博琴 高建宏 编
责任编辑 陈丽
*
*
西安交通大学出版社出版发行
(西安市咸宁西路28号 邮政编码:710049 电话:(029)3268316)
西安电子科技大学印刷厂印装
各地新华书店经销
*
开本 787mm×1 092mm 1/16 印张 20.625 字数:498千字
1999年3月第1版 1999年3月第1次印刷
印数:1~3 000
ISBN7-5605-1016-7/TP·182 定价:25.00元

前　言

近年来，C语言在我国成为各种程序设计语言中最被人们关注的语种之一，计算机专业人员乐意用它写程序，非软件人员也把它视为一种时尚，抱着空前的热情用它开发软件。

人们对它如此厚爱，并非盲从，因为C语言确有许多诱人的地方：其一，C兼有高级语言和汇编语言的优点，描述问题能力强，灵活，应用面宽，编译程序规模小、目标质量高，可移植性好；其二，适合结构化程序设计，顺应目前国内程序设计趋势；其三，独领风骚的UNIX的广泛使用起到了推波助澜的作用，因为UNIX的95%的代码是用C写的。

本书是作者长期从事计算机软件教学工作积累的总结，在构思本书的框架和选材时，我们根据以下两个原则：一是要使本书适合教学需要，兼顾计算机专业和非专业人员的不同目标；二是力图使本书有一个新的结构，而且从内容上真正说清楚"C的特点是什么?"具体地说，是从以下5方面来体现的。

1. 本书的内容组织是从程序在内存运行时存取数据的三种途径(I/O设备、内存本身自带及外存)入手，逐步引入与其对应的输入/出操作，常量、变量及其类型和文件的操作。

2. 指针是C的最重要特色，本书从扩大因子概念出发，圆满地解释了指针与简单变量、指针与数组以及指针与结构之间的关系，特别是指针与二维数组的关系。

3. 以ANSI标准推荐的函数原型法为基础介绍函数的定义与使用。另外介绍了用户自己如何建立函数库，使读者真正体验到标准函数与用户自定义函数的统一，最后从结构化程序设计思想出发总结了函数间数据共享的各种方法及优缺

点。

4. 对 C 的特点，在有关章节中通过例子来分析，使读者有感性的认识。最后再对 C 所以有这些特点的因素进行了归纳，使之认识得以升华。最后一章还介绍了 C 程序的动态调试。

5. 注意到硬件、操作系统与 C 的关系。从 C 对硬件的依赖性出发说明了它的可移植性特色。第 13 章介绍了操作系统对语言的支撑关系。

书上的例子全部在 Turbo C2.0 版上通过，并且对表达式的副作用问题还专门在 386 XENIX System V 及 Sun SPARC station UNIX 环境下验证。

本书是 1993 年出版的《C 语言程序设计》的修订本，主要的变动是：① 对全书的章节和内容重新安排：由 12 章改为 14 章，改写了第 6～8 章，构造数据结构的内容从原书的第 4 章移到第 7 章；② 增加了面向对象技术及 C++ 简介(第 12 章)，对 C 的发展有所交代。

本书仍由冯博琴组织修订，参加修订的有齐勇(第 1～5，12～14 章)、王建仁(第 6～8、10、11 章)、冯博琴(第 9 章)，最后由冯博琴统稿。西安医科大学计算中心主任陈康教授仔细审阅了本书，并提出许多宝贵意见；几年来，本书 1993 年版的许多读者也给我们诸多建议和指正，为这次修订提供了思路，在此一并表示衷心感谢。

由于作者水平有限，修订之中可能仍有疏漏之处，恳望同行和读者不吝指正。

作者

1998.10

目　　录

第 1 章　引　　论

在众多的软件产品中，唯 UNIX 是能够在各种机型、各种档次的计算机上广泛运行的系统，在 1988 年，世界信息产业十大要闻中，“UNIX 风云遍全球”名列榜首。UNIX 系统以其精巧、高效、功能强、移植性好而著称。这种直接建立在硬件环境上、为其它软件提供支撑的系统软件其应用之广在计算机的发展史上是罕见的。它的设计师 Ken Thompson 和 Dennis M. Ritchie 也由此而获得了图灵奖。UNIX 系统的成功一方面是由于设计者对系统的关键部分作了恰如其分的选择与合理的构造，另一方面则要归功于 C 语言了（系统的 95% 是用 C 语言编写），C 语言的使用对系统的可靠性、可移植性起到了决定性作用。

在 PC 机领域中更是如此，很多著名的软件都是用 C 语言编写的，如：FoxBASE，ORACLE，Wordstar，R:base for DOS，Novell Netware，Internet 上的 Web Server/ Browser 等等。

C 语言现在已成为软件产品开发者最主要的编程语言之一，也是第一流专业程序员最常用的语言。C 语言有如此大的魅力，那么它到底是一种什么样的语言呢？它与现在众多的程序设计语言相比到底有哪些不同呢？

首先来回答第一个问题。很多人将 C 语言称为“低级语言的高级形式”或“汇编语言的速记形式”。C 语言的创始者 Ritchie 称 C 是一种通用的、较“低级的”程序设计语言。由此可知，C 语言是一种集高级语言与汇编语言特点于一体的一种中级语言。

1.1　有了高级语言为什么还要引入中级语言——C 语言产生的背景

C 语言是 1972 年 UNIX 系统的设计者之一 Dennis M. Ritchie 为重写 UNIX 系统而发明的，其发展过程为

ALGOL60→CPL→BCPL→B→C

ALGOL 语言比 FORTRAN 晚几年，它比 FORTRAN 更完善，但它由于太抽象没有得到真正的推广，但是 ALGOL 中提出的语法规则和模块结构对后来的程序设计语言有较大的影响（PASCAL 语言就是在这个基础上产生的）。1963 年剑桥大学和伦敦大学为了使 ALGOL 语言得到实用，在此基础上构造了 CPL（Combined Programming Language），后来剑桥大学的 Martin Richards 于 1967 年又对 CPL 进行浓缩，保留较好的基本特性，产生出了 BCPL（Basic Combined Programming Language），它规模小，便于实现和学习，且最大的特点是移植性好，所以在欧洲一些国家仍在使用。

特别值得一提的是，当时计算机中的系统软件都是用汇编语言编写的，尽管当时已经有了高级程序设计语言，但在编写像操作系统这样与硬件密切相关的系统时，高级语言是无能为力的，而使用像汇编这样低级的、非结构化的语言编写较大的系统，可靠性又是无法保证的，比如为 IBM360 机器编写的操作系统，花费 5 000 人年，规模为几百万条指令，但在以后的每个版本中总有上千个错误。其它一些较大的系统也都是如此，这也就是当时所谓的“软件危机”。当时也已萌芽了一些结构化程序设计的思想，60 年代末 N. Wirth 提出了 PASCAL 程序设计

语言,较好地体现了结构化程序设计的一些原则,但是由于高级语言就是让用户摆脱对硬件环境的依赖,因而对操作系统这样的软件,高级语言是无法实现的。长期从事系统软件设计的Ken Thompson 和 Dennis M. Ritchie 谙知这一矛盾,他们试 图寻求一种解决的方法,因此,Ken Thompson 在 1970 年又对 BCPL 进一步优化,在 PDP-11/20 上实现了一个 B 语言(取其 BCPL 的第一个字母),并用它重又编写了 UNIX 操作系统及大部分的实用程序。但是 B 语言是一种无类型面向机器字的语言,所以在描述各种数据结构时是很困难的。另外,由于 B 语言最后产生的是解释执行的代码,运行速度较慢。

鉴于这一原因,1972 年 Dennis M. Ritchie 又在 B 语言的基础上增加了许多数据类型,并为其编写了一个实用的编译程序,取 BCPL 的第二个字母,称为 C 语言。随之用 C 语言重新编写了 UNIX 系统,从此使 UNIX 产生了强大的生命力。由此可知,C 语言是中级语言恰恰是它的长处所在,它正好解决了用高级语言实现系统软件困难,而用汇编语言实现时可靠性及移植性都差的这一矛盾。

1.2 C语言的特点

C语言的特点就在于它将低级语言与高级语言的特点集于一体。具体表现在三个方面:

(1) 具有汇编语言的高效及对计算机中基本成份的处理能力。

用 C 语言可以编写出效率几乎接近于汇编语言代码的程序。据统计用 C 语言编写程序所产生的代码与汇编语言的代码比例为 1.2:1。体现在 C 语言中就是它提供了高效的 + + 及 - - 运算,用指针处理数组以及用零和非零作为逻辑值及寄存器变量,充分利用这些因素可以编写出高效的程序。

C语言可以像汇编语言那样对位、字节和地址这些计算机中的基本成份进行操作,还可以通过转义字符对控制字符进行处理。这些是编写与硬件密切相关的系统软件所必备的条件,其它高级语言在这方面是望尘莫及的。

(2) 具有可移植及结构化的特点。

可移植性也是一般高级语言的一个特点,但是 C 语言在这方面却有其独特之处。C 语言本身很小,关键字只有 32 个。它将所有与外部设备有关的控制部分都抛给了库函数,而 C 编译程序本身仅处理一些与硬件关系不密切的有关数据类型及程序的流程控制问题。这样一来,只要保证不同机器及操作系统下的 C 语言库函数接口一致,那么 C 语言程序就可以很容易地移植到不同的机器上。

C语言提供的丰富的流程控制语句及函数的外部结构,使得它完全可以按结构化程序设计的思想编写程序。

(3) 简洁灵活。

C语言提供了丰富的运算符及表达式,特别是其赋值表达式、条件表达式、指针运算以及使用零和非零作为逻辑值,从而可使程序员写出很简炼的程序。C 语言的灵活性体现在它很少限制、很少强求。它不是像 PASCAL 那样强类型的语言。它允许几乎所有的类型转换。它对于数组的边界及变量的地址越界问题是留给程序员自己处理的。

由 C 语言的这些特点可以看出,它可以代替汇编语言,编写的程序可以产生出几乎接近汇编语言的高效代码,而同时又具有高级语言的结构,这也就是它的魅力所在。

C语言在不同机器上、不同操作系统环境下版本很多。多年来,C 语言的发明者 Dennis M. Ritchie 与 Brain Kernighan 合写的 *The C Programming Language* 一书一直被作为 C 语言的公认标准,简称 K&R C。随着微机的普及,C 语言被广泛地采用,由于没有一个统一的标准,因而推广过程中都会根据具体机器的特性作一些增舍,特别是库函数部分。这导致了 C 语言移植上的一定困难。于是,1983 年 ANSI 设立了一个委员会花了近五年的时间,兼顾到 DOS 操作系统单用户的环境,在 K&R C 基础上经过修订发表了 ANSI C 标准。1990 年国际标准化组织 ISO 在此基础上颁布了 C 标准(ISO9899—1990)。现在比较新的 C 语言版本都支持这一标准。

第 2 章　程序运行的基本过程及C语言程序的基本结构

本章内容是学好一门程序设计语言的基础。首先给出输入输出的概念并通过一般程序的运行过程归纳出程序对数据存取的途径；另外，通过一些简单的例子使读者对 C 语言的程序结构有一个大概了解。这些对于后续章节内容的掌握很有帮助。

2.1　输入输出概念

计算机是由 CPU、内存储器(简称内存)、外存储器(亦称辅助存储器，简称外存，像磁盘、磁带等)、用于输入输出的外部设备(键盘、显示器、鼠标器、打印机等)以及其它控制部件组成。而输入输出是相对内存讲的，即计算机的内存与外存及外部设备之间进行数据传输的过程。将数据从外存或外部设备(键盘等)送往内存的过程称为输入；反之，将数据从内存送往外存或外部设备(显示器、打印机等)的过程称为输出。如图 2-1 所示。由于外部设备主要是用于输入输出，所以亦称之为输入输出设备，简称为 I/O 设备(Input/ Output)或外设，另外，亦将显示器与键盘称为终端设备。

图 2-1　输入输出的含义

2.2　程序的运行过程及程序中存取数据的途径

任何程序，只要不是用机器指令编写的，在运行前都必须被转换成相应的机器代码指令序列(由编译程序或解释程序转换)，最后装入内存(由操作系统装入，解释程序例外)，再由 CPU 执行其指令序列。如果我们不考虑程序的细节，而只从外部来看，可以将程序看成是一个函数 F()，它接收原始数据集合 X，经过处理运算，最后产生出结果数据集合 Y。即

$$Y = F(X)$$

程序是在内存中运行，它要处理的数据 X 将从何处来呢？由上一节可知，X 可来自三个方面：

(1) 来自于输入设备，即在运行中程序等待用户从键盘等设备输入要处理的数据(用 X1 表示)。

(2) 来自于外存。要求处理的数据可以由另外的程序产生好或通过其它方法(如编辑工具)将它们以文件的方式存放到外存上(用 X2 表示)。

(3) 程序本身自带。数据直接以常数或初值的方式随程序一同装入内存(用 X3 表示)。程序可直接对其处理。

对于数据 Y 也类似：

(1) 送往输出设备,比如显示器、打印机等,将结果出示给用户(用 Y1 表示)。

(2) 送往外存。作为结果保存或作为其它程序所需的数据(用 Y2 表示)。

至于如何实现这些数据的传输,则是第 3,4,9,10 章的内容。图 2-2 给出了程序对数据的存取途径:

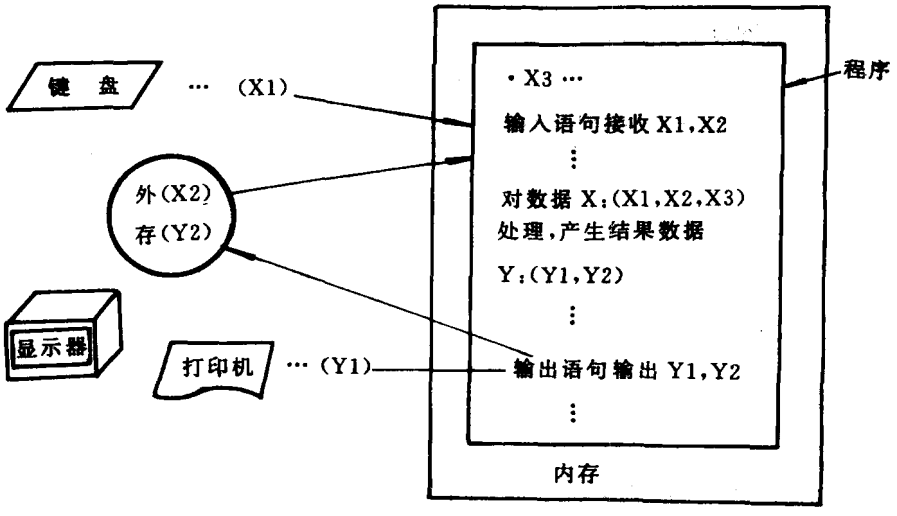


图 2-2　程序对数据的存取途径

2.3　C 程序的基本结构

C 语言程序是一种函数结构,它是由一个主函数 main()和若干个子函数组成。一个 C 程序可以没有其它子函数,但必须要有一个主函数。C 程序的一般形式如图 2-3 所示:

```
全局量说明
main()
{
        局部量说明;
        语句序列;
}
sub1()
{
        局部量说明;
        语句序列;
}
sub2()
{
        局部量说明;
        语句序列;
}
        …
subN()
{
        局部量说明;
        语句序列;
}
```

图 2-3　C 程序的一般形式

图中的main(),sub1(),…,subN()是由用户自己定义的。全局量的说明可以没有,也可以放在任意两个函数之间。函数体是由"{"及"}"括起来的部分,不能少(相当于PASCAL中的BEGIN与END)。C程序中的函数将其它高级语言中的函数子程序(PASCAL,FORTRAN中)及过程子程序(BASIC, PASCAL,FORTRAN中)统一起来,给编写者提供了统一形式,当函数的执行无返回值时,它就相当于过程子程序;当它有返回值时就相当于函数子程序。

例 2-1 用不同的方法在显示器上显示:"C Programming Language"和"C语言程序设计"。

方法 1:

```
main()
{   printf("C programming Language \n");
    printf("C 语言程序设计 \n");
}
```

方法 2:

```
main()
{   print _ English();
    print _ Chinese();
}
print _ English();
{
    printf("C Programming Language \n");
}
print _ Chinese();
{
    printf("C 语言程序设计 \n");
}
```

程序中的print _ English()与print _ Chinese()是两个无返回值的函数,相当于其它语言中的过程子程序,而主函数中仅调用这两个函数执行。从方法2可以看出一个大程序的缩影,这种函数结构可使用户将一个复杂的问题分解成若干个相对独立的简单的子问题,并分别由不同的函数实现,以达到结构化程序设计的目的。

程序中的printf()为一个标准函数,它实现将引号括起来的内容原样输出,符号"\n"表示回车字符,不加此符号时,输出字符串后光标停在字符串的末尾。C语言不提供输入输出语句,而是通过调用标准函数实现输入输出操作。C程序的书写格式很灵活,各语句之间用分号";"隔开,多个语句可以写在一行上,也可各占一行。当然为了程序的易读性,最好还是各占一行,因为这样并不增加可执行程序的长度。

C语言的这种函数块结构可以由不同的用户编写,即根据需要将C语言的若干函数由不同的人编写,并且每个人编写的函数都作为源程序文件存放,经过分别编译成功后,最后连接装配成一个可执行的程序。这对于由几个人合作编写一个大程序带来了方便。详见第4章中C语言的上机过程。

2.4 注 释

C语言中的注释是以/* 打头,后跟任意的字符序列,最后以 */结束,它相当于BASIC的